

# **Proceso para el desarrollo de emuladores de unidades electrónicas de control basado en la técnica HIL- Review**

**Freddy Alexander Díaz González**

Pontificia Universidad Javeriana, Bogota, Colombia, diaz.freddy@javeriana.edu.co

**Darío Amaya Hurtado**

Universidad Militar Nueva Granada, Bogota, Colombia, dario.amaya@unimilitar.edu.co

**Diego Alejandro Patiño Guevara**

Pontificia Universidad Javeriana, Bogota, Colombia, patino-d@javeriana.edu.co

## **RESUMEN**

Los sistemas de manufactura han sido una parte fundamental para el desarrollo económico de los países, adicionalmente, con la evolución de la electrónica digital se posibilita la comunicación y control en red, de los sistemas de manufactura, para lo cual el desarrollo de las unidades electrónicas de control, requiere de formulación de modelos basados en comportamiento que contemplen no solamente la dinámica sino también los eventos que afectan el sistema híbrido. En la validación de controladores para sistemas de dinámica híbrida se requieren de emuladores para evitar sobre costos y daños en las plantas de los sistemas de manufactura, la técnica HIL es una herramienta fundamental para la rápida implementación de emuladores. En este artículo se hace un barrido de los diferentes modelos de desarrollo de unidades electrónicas de control y las herramientas software y hardware empleados en los modelos.

**Palabras claves:** Eventos Discretos, Sistemas Híbridos, Técnica Hardware In the Loop.

## **ABSTRACT**

Manufacturing systems have been fundamental to the economic development of the country, in addition, with the evolution of digital electronics enables communication and network control of manufacturing systems, for which the development of the electronic units control, requires formulation of models based on behavior that include not only the dynamics but also the events that affect the hybrid system. For validation of controllers for hybrid dynamic systems require emulators to avoid cost and damage to plants manufacturing systems, the HIL technique is an essential tool for the rapid deployment emulators. This article makes a sweep of different development models electronic control units and software and hardware tools used in the models.

**Keywords:** Discrete Event, Hybrid Systems, Technical Hardware In the Loop.

## **Introducción**

Los sistemas de manufactura han sido una parte fundamental para el desarrollo económico de los países (Wang & Shen, 2007), con el desarrollo de la electrónica digital se posibilitó la comunicación y control en red de los sistemas de manufactura, lo que dio origen a los Sistemas de Manufactura Integrada por Computador (CIM) (Aardal, 1995) (Norman, 2000). La implementación de sistemas CIM favorece los actuales modelos de fabricación de productos, los cuales buscan una salida rápida al mercado combinada con la constante actualización y cambio de los componentes con que se fabrican. Esta dinámica genera el ajuste de los procesos de

manufactura, para viabilizar la producción de las nuevas piezas de los nuevos productos, esto complementa los Sistemas de Manufactura Flexible (FMS) (Rezaie, 2009).

Un reto importante de los FMS es la reducción de los ciclos de vida de los productos y la dimensión de la producción. Esto implica la necesidad de incorporar, rediseñar o reprogramar los componentes de los CIM, esto es una tarea compleja que puede tomar un tiempo considerable (Rocha, 1994), adicionalmente, la incorporación de nuevos módulos al CIM, implica el desarrollo nuevos sistemas de control, para lo cual el proceso de diseño se basa en el modelo del sistema. Son varias los formalismos que guían el proceso de modelamiento los sistemas de manufactura, entre los más utilizadas están los autómatas finitos y las redes de Petri, Adicionalmente, la mayoría de los procesos industriales combinan dinámicas continuas con dinámicas basadas en eventos, para los que los formalismos tradicionales no son suficientes (Lygeros & Johansson, 2003).

### **Sistemas dinámicos de eventos discretos**

Durante las últimas décadas, la rápida evolución de las tecnologías de computación, comunicación y sensores; ha traído consigo la proliferación de nuevos sistemas dinámicos de gran complejidad (Wainer, 2009); algunos ejemplos son: las redes informáticas y de comunicación, sistemas de fabricación automatizados, sistemas de control aéreo (Wainer, 2009). Una parte importante de la actividad en estos sistemas, se rige por normas operativas establecidas por los seres humanos o creados por estos, por lo cual su dinámica se caracteriza por los sucesos asíncronos de eventos discretos, algunos controlados, como oprimir una tecla o el envío de un paquete de mensajes; y otros no, como un fallo espontaneo de un equipo o una pérdida de paquetes en una cola de datos; algunos detectados por los sensores y otros no (Cassandras, Introduction to Discrete Event Systems Second Edition, 2008). Los sistemas dinámicos con estas características denominan Sistemas de Eventos Discretos (DES) (Zeigler, Theory of Modeling and Simulation Second Edition, 2000). En general los DES son sistemas dinámicos que evolucionan de acuerdo con la ocurrencia abrupta de eventos físicos, en intervalos irregulares de tiempo o asíncronos (Ramadge, Some Tractable Supervisory Control Problems for Discrete-Event Systems Modeled by Buchi Automata, 1989).

### **Concepto del evento en los Sistemas de Eventos Discretos**

Para DES, el concepto del evento es visto como algo que ocurre instantáneamente, que hace una transición de su valor, de un estado a otro (Zeigler, Theory of Modeling and Simulation Second Edition, 2000). Un evento puede ser identificado con una acción específica tomada, puede ser visto como una ocurrencia espontánea dictado por la naturaleza del sistema o puede ser el resultado de varias condiciones que de repente se reunieron (Ramadge, The Control of Discrete Event Systems, 1989). En un DES el cambio de estado no se produce de forma síncrona o predecible en el tiempo (Hrúz & Zhou, 2007 ).

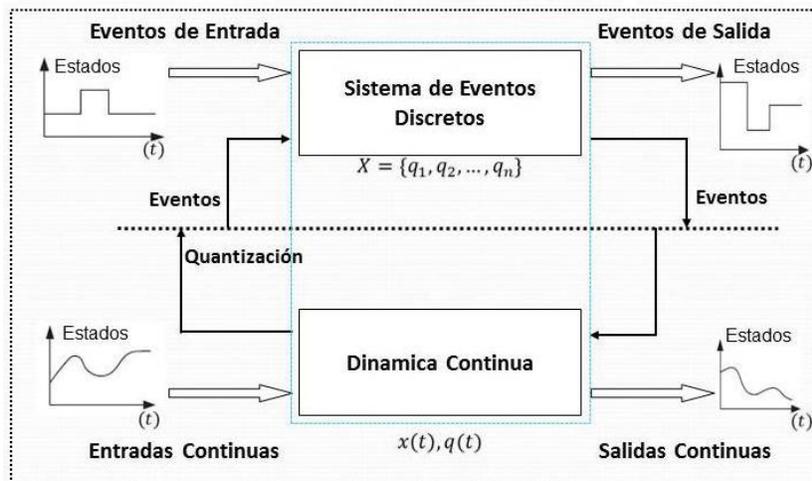
Para relacionar el concepto de eventos discretos con el modelamiento de sistemas, Bernard Zeigler propuso un formalismo general para la representación DES. Este formalismo es denominado “Discrete Event system Specification” (DEVS), este se extiende al caso continuo, facilitando así la combinación de modelado discreto/continuo. DEVS ha sido implementado en una variedad de entornos de simulación, como extensiones de los diversos lenguajes de programación orientados a objetos resaltando CLOS y C++ (Zeigler, Formalism and Methodology Unity of Conception Diversity of Application, 1993).

En 2007 Hrúz y Zhou proponen un formalismo de modelamiento para sistemas híbridos que llamaron “Discrete-Event Dynamic Systems” (DEDS), en este se define como se toman los valores del sistema en el transcurso del tiempo ( $Q_1, Q_2, Q_3 \dots Q_n$ ), también define los eventos que ocurren en el tiempo y los impulsos o señales, que estimulan la evolución del sistema ( $e_1, e_2, e_3, e_4 \dots e_n$ ). (Hrúz & Zhou, 2007 ).

### **Sistemas híbridos**

Un sistema híbrido consiste en la combinación de valores de estado continuos y de eventos, de combinaciones de variables de tiempo continuo y discreto, ya que a través de ellos se produce la interacción con otros sistemas del medio, los cuales influyen en el comportamiento de la dinámica del sistema en estudio (Antesaklis, 2000, Julio).

En la figura 1 se muestra la estructura de un sistema de arquitectura híbrida, en la cual interactúan los componentes en función de eventos, con los componentes en dominio en tiempo continuo. El modelo que se define como dinámica continua, puede estar representado por ecuaciones diferenciales o ecuaciones en diferencia, en otras palabras, para los análisis y diseños de sistemas Linear Time-Invariant (LTI), la transformada Z representa el mismo rol que la transformada de Laplace (Oppenheim, 1998) (Proakis, 2007). En un sistema híbrido el concepto de discreto se analiza desde el punto de vista del evento o suceso asíncrono (Cassandras, Introduction to Discrete Event Systems Second Edition, 2008) (Zeigler, Theory of Modeling and Simulation Second Edition, 2000) (Ramadge, Some Tractable Supervisory Control Problems for Discrete-Event Systems Modeled by Buchi Automata, 1989).



**Figura 1. Arquitectura básica de un sistema híbrido**

Los sistemas híbridos abarcan un gran rango de aplicaciones que van desde control de procesos de manufactura, hasta sistemas de energía para aplicaciones aeroespaciales. Los sistemas híbridos fueron desarrollados para el estudio de los procesos físicos que exhibe poca suavidad en su comportamiento (LUNZE, 2009).

Para el modelamiento de sistemas híbridos se utilizan formalismos matemáticos de fácil entendimiento, necesarios para las investigaciones en ingeniería de control, matemáticas e informática (Wainer, 2009) (Pepyne, 2000), actualmente existen tres modelos básicos para el estudio de sistemas de control híbridos: el modelo basado en ecuaciones, los modelos gráficos basado en máquinas de estado finitas y los modelos basados en las redes de Petri (Khan, 2007).

### **Modelamiento y control de sistemas de arquitectura híbrida**

En 1966 iniciaron los estudios de los sistemas dinámicos en tiempo discreto con la teoría del Autómata Celular (CA) (Von Neumann, 1966); luego en la década del sesenta se genera el modelo matemático de propósito general para describir las relaciones existentes entre las condiciones y eventos, denominado Redes De Petri (David & Alla, 1987); (Alury & Courcoubetisz, 1995) propone el modelamiento de un sistema híbrido basado en un Autómata Finito; (Bail & Alla, 1991) y (David & Alla, 2010) utilizan el concepto de redes de Petri para el modelamiento de sistemas de arquitectura híbrida.

En 1972 Zeigler desarrolla el formalismo para la descripción de sistemas de eventos discretos (DEVs) (Palaniappan, 2006), este formalismo descomponen los sistemas en modelos atómicos que representa la unidad molecular de procesamiento, la cual es la pieza fundamental y más básica de los modelos DEVs (Zeigler, Today-

recent advances in discrete event-based information technology, 2003) (Bergero & Kofman, 2008). Los modelos acoplados del formalismo DEVS pueden convertirse en componentes de sistemas más grandes, lo cual lleva a la descomposición y construcción jerárquicas.

Basado en el formalismo DEVS (Nikolaidou & Dalakas, 2008) desarrolló un lenguaje gráfico de propósito general de modelado UML modificado, al cual denominó: El Lenguaje de Modelado de Sistemas (SysML), este se ha utilizado para aplicaciones de sistemas de ingeniería. Es compatible con la especificación, análisis, diseño, verificación y validación de una amplia gama de sistemas. Estos sistemas pueden incluir hardware, software, información, procesos, personal e instalaciones. SysML v.1.0 fue adoptada por el Object Management Group (OMG) como estándar en julio de 2006 (the Object Management Group, Inc. (OMG), 2006).

Son varios los trabajos que se ha realizado empleando el formalismo DEVS; Capocchi (Capocchi, Bernardi, Federici, & Bisgambiglia, 2003) presenta una propuesta de un método de transformación de los modelos desarrollados en VHDL para modelos DEVS acoplados. Como validación del método se realizó un modelamiento y simulación de un registro de ocho bits, dejando abierta la posibilidad de desarrollar un modelo integrado y un entorno de simulación para esta tarea de conversión de modelos. Gentili (De Gentili, 2005) describe el modelamiento de un proceso de fabricación de productos lácteos utilizando el formalismo DEVS. Como el modelo propuesto, consigue separar los modelos del sistema de control del modelo de datos, proporcionando flexibilidad y portabilidad en el desarrollo del modelo general del proceso, destacando la facilidad del modelo DEVS acoplado, para permitir representar en unidades el modelo de un sistema complejo. HU (Hu, Ganapathy, & Zeigler, 2006) presenta un trabajo focalizado en la robótica colaborativa basado en un proyecto incremental, utilizando la simulación como primer paso para la evolución gradual de la implementación sobre dos robots. Los entornos de simulación están basados en el formalismo DEVS, con resultados cuantitativos de las simulaciones en las validaciones. Este demuestra la facilidad de la implementación de este modelo del proyecto. Orooji (Orooji, Sarjoughian, & Taghiyareh, 2010) realiza un modelamiento y simulación de un sistema basado en el concepto de multi-agentes de una plataforma de ayuda académica, que ofrece a los alumnos una asesoría de los cursos ofertados que pueden tomar, al momento de realizar sus inscripciones, esta permite validar sus propuestas con un posterior modelamiento y simulación, antes de ser implementadas.

Tradicionalmente en la implementación de sistemas de control con dominio en el tiempo, se han utilizado los sistemas digitales para cuantificar periódicamente las señales entregadas por los sensores, generalmente por medio de un conversor análogo a digital (A/D) (Astrom & Wittenmark, 1997). Sin embargo, las técnicas de conversión A/D realizan un muestreo sincrónico dejando la planta sin control entre las muestras sucesivas. Este hecho suele degradar la respuesta dinámica y afectar las regiones de atracción en sistemas no lineales (Kofman, Discrete Event Based Simulation and Control of Continuous Systems, 2003). Existen algunas aplicaciones donde el muestreo está basado en eventos. Se pueden encontrar ejemplos en encoders de rueda y acelerómetros, estos proporcionan trenes de pulsos en lugar de las señales de medición continua, adicionalmente los sistemas de colas son otra aplicación donde las señales se actualizan de forma basada en eventos, lugar de forma continua (Anton, 2007).

Dormido (Dormido, 2008) expone varias aplicaciones del control basado en eventos, como actuadores adaptados, redes de sensores inteligentes, sistemas de modulación IPFM, sistemas de supervisión de procesos, aplicaciones basadas en colas y buffers. También resalta que son pocas las aplicaciones de control basado en eventos en los niveles inferiores de la jerarquía de control en aplicaciones industriales, pero resalta las aplicaciones de control por eventos en los niveles de supervisión de las líneas de producción.

En el avance de la teoría de control basado en eventos, se ha llegado hasta el reemplazo del método matemático para el modelamiento de las plantas, pasando de las ecuaciones en diferencias a la cuantificación de estados, proponiendo una alternativa al muestreo periódico de las señales de sensores, por eventos producidos por cambios de nivel. Zeigler (Zeigler, Theory of Modeling and Simulation Second Edition, 2000) presenta un método para simular sistemas de tiempo continuo a través de cuantificación de estados, proponiendo discretizar las variables de estado, basado en eventos producidos por los cambios de la señal y no por un periodo de muestreo. Continuando con la teoría de cuantificación de estados, Kofman (Kofman & Junco, Quantized-state systems: a DEVS Approach for continuous system simulation, 2001 ) desarrolla un método integrador que denomina sistemas de

estados cuantificados (QSS); luego basado en QSS propone un sistema de control por estados cuantificados (QSC), este método propone la implementación de controladores digitales, que son inicialmente proyectados como controladores continuos, para luego cuantificar sus variables de estado con QSS, este tipo de control digital es descrito como un sistema de eventos discretos basado en un sistema de muestreo asíncrono (Kofman, Quantized-state control: a method for discrete event control of continuous systems, 2003).

### **Herramientas de software para la simulación de sistemas híbridos**

La alta complejidad de los procesos de manufactura requiere herramientas de software para ayudar a su análisis, modelamiento, simulación y control (Wei, 2007). Tradicionalmente los simuladores (como Simulink I (MathWorks, 2001)), emplean un mecanismo de ejecución en el dominio del tiempo, lo cual es totalmente contrario a lo requerido por los sistemas de eventos discretos (DES). SimEvents (MathWorks, 2005), está diseñado para simular DES, incorporando en Simulink. SimEvents habilita funcionalidades que permiten a una efectiva co-existencia de los componentes en el dominio del tiempo y los del dominio de los eventos, que conforman los sistemas híbridos de manufactura (Cassandras, Hybrid System Simulation With Simevents, 2006).

SimEvents contiene las bibliotecas y conjuntos de bloques de tipo colas, servidores, puertas e interruptores, que permiten implementar un modelo con los componentes básicos de un DES. Por interconexión de estos componentes, se puede modelar una red de comunicaciones o un sistema de manufactura de tipo DES. SimEvents también puede ser utilizado conjuntamente con Simulink y facilitar el diseño simulación del sistemas híbridos (Wei, 2007).

El uso de herramientas software como SimEvents y Simulink para el modelamiento de sistemas de manufactura es tema de interés de los investigadores del área de automatización industrial; Nica (Nica, 2008) modela las colas de materiales que alimentan diferentes procesos, para definir los espacios de almacenamiento utilizando MATLAB, SimEvents. Gawthrop (Gawthrop, 2008) desarrolló una estrategia de control implementada en HIL enfocado a emulación, con la cual resuelve problemas de estabilidad ocasionadas por la dinámica no deseada de la función de transferencia, el modelamiento y la implementación rápida; lo realizó con ayuda de la herramienta software MATLAB, Simulink. Canale (Canale, 2009) implementa un controlador en un dispositivo embebido utilizando, la técnica de Software-In-the-Loop (SIL), para realizar las pruebas y simulación del sistema utilizó Simulink. Zhu (Zhu, 2009) utilizan el concepto de HIL para implementar un sistema de un vehículo eléctrico, el control lo implementa en un computador de escritorio utilizando Matlab, Simulink, la comunicación con el sistemas la hace con un bus CAN.

En los trabajos realizados por (Hae, 2005) y (Furfaro, 2009) se utiliza una extensión de la metodología DEVS denominada "Real-Time Discrete Event System Specification" (RT\_DEVS), con la cual embeben sistemas de control de tiempo real en micro procesadores, pero no relacionan la comunicación del controlador. (Michael, 2008) y (Isermann, 2007) presentan modelos para el desarrollo de unidades electrónicas de control de sistemas mecatrónicos, utilizados en la industria automotriz, el primero se aleja del modelo tradicional tipo "V" y el segundo lo aplica partiendo desde los requerimientos funcionales para el diseño hasta llegar a la integración de sistemas para control de la producción; los dos contemplan las ventajas del uso de herramientas software y hardware para la construcción rápida de sistemas de control embebidos, pero ninguno contempla la naturaleza híbrida de los sistemas de producción. Grepl (Grepl, 2011) presenta un trabajo de implementación de sistemas de control de tiempo real utilizando las herramientas de Matlab (Simulink). En este trabajo se propone un método para la validación de los modelos de plantas, embebiéndolo en un sistema de tiempo real, utilizando herramientas de generación automática de código, adicionalmente valida el modelo del controlador implementándolo en sistemas Hardware/Software comerciales y conectándolo a la planta real. Sin embargo, Grepl no menciona el uso de herramientas que contemplen la dinámica de eventos discretos, con la cuales se contemplan las características híbridas de los sistemas de producción.

### **Descripción de la técnica Hardware In the Loop**

En la actualidad el desarrollo de los sistemas de control embebidos con aplicaciones que requieren de alta robustez, como lo son equipos médicos, maquinaria industrial, sistemas de generación de energía, desarrollos aeroespaciales e industria automotriz; basa sus validaciones iniciales en pruebas realizadas sobre emuladores electrónicos (Jovanov & LeSueur, 2009), con el fin de realizar análisis de viabilidad y repetitividad, identificar casos de fallos y establecer los procedimientos de mantenimiento y soporte (Ryssel, Ploennigs, Kabitzsch, & Folie), todo esto acelera los tiempos y disminuye los costos del proceso para el desarrollo de sistemas de control embebidos (Winkler & Gühmann).

El uso de emuladores en la validación de los prototipos de los sistemas de control, se soporta en la implementación del modelo basado en el comportamiento de la planta que se desea controlar, en una unidad electrónica (Kramer, Mishra, Brauneis, & Schmidt, 2012); el emulador debe responder a las señales de entrada como lo haría la planta real (Azarnasab, Peiman, & Behrouz). Esto posibilita el análisis de las acciones de control ante los diferentes tipos de estímulos y perturbaciones, incluyendo los casos extremos de fallo, que puedan afectar al sistema en su funcionamiento en campo. El emulador evita el desgaste y daño de las plantas, durante la ejecución de las pruebas de validación (Winkler & Gühmann).

El proceso de identificar el modelo de una planta que se desea controlar, desarrollar el controlador para esa planta, implementar el modelo de la planta sobre un sistema electrónico y finalmente llevar a cabo un protocolo de pruebas del controlador desarrollado sobre modelo implementado, se define como aplicación de la técnica Hardware In the Loop (HIL) (Grega, 1999). El éxito de HIL se basa en la correcta implementación del modelo en un sistema electrónico, para esto generalmente se usan herramientas que garanticen los tiempos de repuesta, como lo son los Sistemas Operativos de Tiempo Real y computadores específicos que puedan recibir y generar las señales eléctricas idénticas a las relacionadas con la planta real. Con HIL la validación previa del controlador antes de interactuar con la planta, puede superar a los análisis extraídos de una simulación, disminuyendo los riesgos propios del proceso completo de desarrollo (Jung & Tsiotras).

### **Herramientas Hardware Real Time específicas para Implementaciones HIL**

Existen en el mercado varios proveedores de soluciones de hardware en tiempo real para la implementación de emuladores HIL, entre otros National Instruments, Opal-RT, dSpace, ETAS, Speedgoat y Quanser.

Los productos ofrecidos por dichas compañías pueden ser configurados de manera tal que se adapten a las necesidades del consumidor, ya que la mayoría son modulares. Estos sistemas tienen un alto costo debido a sus prestaciones. En la tabla 1, se pueden observar algunos productos de los proveedores antes mencionados:

**Tabla 1: Productos que se encuentran en el mercado para hacer Hardware In the Loop**

<b>PRODUCTO Y FABRICANTE</b>	<b>PROCESADOR DE TIEMPO REAL</b>	<b>MÓDULO I/O</b>	<b>APLICACIONES</b>	<b>Link de consulta</b>
Sistema de referencia general para simulador HIL  - Fabricante: National Instruments	Procesador Intel Celeron de 2.0 GHz, 1GB RAM, 80 GB en Disco Duro	FPGA Virtex-5 conectada a módulos I/O. (USD\$3080) 8 entradas análogas 16 Bits, 0V-10V , 8 salidas análogas, 96 E/S digitales	Aeronáutica, Automóviles.	<a href="http://sine.ni.com/nips/cds/print/p/lang/es/nid/207822">http://sine.ni.com/nips/cds/print/p/lang/es/nid/207822</a>
Sistema para simulación en tiempo real en HIL  - Fabricante: ETAS	Procesador Freescale MPC755 a 400MHz, 40MB de RAM.	Módulos DAC, ADC, PWM, independientes cada uno	Automóviles	<a href="http://www.etas.com/en/downloadcenter/11340.php">http://www.etas.com/en/downloadcenter/11340.php</a>
Sistema para prototipado de control y pruebas en HIL.  - Fabricante: OPAL-RT	Intel Core 2 Duo 2.4 GHz	Basado en FPGA Virtex-2, 64 líneas de E/S	Sistemas eléctricos HVDC, Automóviles, Aeronáutica, defensa.	<a href="http://www.opal-rt.com/product/rt-accelerator-xpc-target">http://www.opal-rt.com/product/rt-accelerator-xpc-target</a>
Simulador mediano.  - Fabricante: dSpace	AMD Opteron Quad Core 2.9 GHz, 1GB de RAM	16 canales A/D 14 bits, 0V-60V, 20 canales D/A 0V-12V	Automóviles	<a href="http://www.dspace.com/en/pub/home/products/hw/singbord.cfm">http://www.dspace.com/en/pub/home/products/hw/singbord.cfm</a>
Modular Real-Time target machine  - Fabricante: SpeedGoat	Intel Pentium-M 1.8GHz, 1GB de RAM, 1GB de Disco Duro	16 entradas 16 Bits, 8 Salidas análogas 0V-10V, 8 salidas digitales, 8 entradas digitales.	Automóviles, Aeronáutica y defensa, Electrónica, Academia, Dispositivos médicos	<a href="http://www.speedgoat.ch/Products/Real-time-target-machines-Modular.aspx">http://www.speedgoat.ch/Products/Real-time-target-machines-Modular.aspx</a>

Q8 High-Performance H.I.L. Control Board.  - Fabricante: QUANSER	Tarjeta con interfaz PCI	E/S digitales 38, 2 PWM, 8 Entradas analógicas 14 Bits 0V-10V, 8 salidas analógicas	Academia	<a href="http://www.quanser.com/english/html/solutions/fs_q8.html">http://www.quanser.com/english/html/solutions/fs_q8.html</a>
--	--------------------------	---	----------	---

### Conclusiones y perspectivas futuras

El desarrollo de unidades electrónicas de control es parte fundamental en la actualización contante de los procesos de manufactura, por lo cual es necesario utilizar las herramientas de alto nivel para acelerar la validación de los prototipos desarrollados.

Las técnicas de modelamiento de sistemas deben contemplar no solamente el cambio en la dinámica de los componentes, sino que deben ser incluidos los eventos que afecten el comportamiento del sistema, como lo pueden ser la recepción de mensajes o una señal de botón pulsado.

Aunque la técnica para el desarrollo de emuladores HIL es ampliamente usada en la validación de componentes de proyectos de gran envergadura como los sistemas para vehículos, los costos comerciales de las herramientas software y hardware, son demasiado altos como para ser empleadas en el desarrollos de sistemas de control em medianas y pequeñas empresas.

Como trabajo futuro se propone proponer un modelo de desarrollo de unidades electrónicas de control, el cual no requiera la adquisición de herramientas hardware de alto costo que pueden estar sobredimensionadas para aplicaciones sencillas requeridas en medianas y pequeñas empresas

### Bibliografía

- Kramer, V., Mishra, R., Brauneis, P., & Schmidt, K. (2012). *Utilization of a hardware-in-the-loop-system for controlling the speed of an eddy current brake*. Germany: IOPscience.
- Aardal, M. (1995). A Model of CIM. *Advanced Semiconductor Manufacturing Conference*, 158.
- Alury, R., & Courcoubetisz, C. (1995). The Algorithmic Analysis of Hybrid Systems. *Theoretical Computer Science*, 138.
- Antesaklis, P. (2000, Julio). Special Issue on Hybrid Systems: Theory and Applications A Brief Introduction to the Theory and Applications of Hybrid Systems. *PROCEEDINGS OF THE IEEE, VOL. 88, NO. 7*, 887.
- Anton, C. (2007). On Limit Cycles in Event-Based Control Systems . *IEEE Conference on Decision and Control*, 3195.
- Astrom, K. J., & Wittenmark, B. (1997). *Computer-Controlled Systems, third edition*. . New Jersey, USA: Prentice Hall.
- Azarnasab, E., Peiman, A., & Behrouz, F. B. (s.f.). *HARDWARE IN THE LOOP: A DEVELOPMENT STRATEGY FOR SOFTWARE RADIO*.
- Bail, L., & Alla, H. a. (1991). Hybrid Petri nets . *1st Int. European Control Conference, ECC91*, 1472–1477.
- Bergero, F., & Kofman, E. (2008). Desarrollo de un simulador de sistemas hidricos en tiempo real. *XXI Congreso Argentino de Control Automático*, 8.

- Canale, M. (2009). Vehicle lateral stability control via approximated NMPC: real-time implementation and software-in-the-loop test. *IEEE Conference on Decision and Control*, 4601.
- Capocchi, L., Bernardi, F., Federici, D., & Bisgambiglia, P. (2003). Transformation of VHDL descriptions into DEVS models for fault modeling. *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, 1205 - 1210 vol.2 .
- Cassandras, C. (2006). Hybrid System Simulation With Simevents. *Analysis and Design of Hybrid Systems*, 269.
- Cassandras, C. (2008). *Introduction to Discrete Event Systems Second Edition*. Boston: Springer Science+Business Media, LLC.
- David, R., & Alla, H. (1987). Continuous Petri nets. *In 8th European Workshop on Application and Theory of Petri Nets*.
- David, R., & Alla, H. (2010). *Discrete, Continuous, and Hybrid Petri Nets*. Berlin, Alemania: Springer-Verlag.
- De Gentili, E. (2005). Devs and Fuzzy logic to model and simulate a manufacturing process. *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, 557 - 564 .
- Dormido, S. (2008). Muestreo, control y comunicaciones basados en eventos . *Revista Iberoamericana de Automatica e Informatica*, 26.
- Furfaro, A. (2009). A development methodology for embedded systems based on RT-DEVS. *Springer*, 127.
- Gawthrop, P. (2008). Emulator-based control for actuator-based hardware-in-the-loop testing. *ScienceDirect*, 908.
- Grega, W. (1999). *Hardware-in-the-loop simulation and its application in control education*. San Juan, Puerto Rico: Department of Automatics, University of Mining and Metallurgy.
- Grepl, R. (2011). Real-Time Control Prototyping in MATLAB/Simulink: review of tools for research and education in mechatronics. *IEEE Intemational Conference on Mechatronics*, 886.
- Hae, S. (2005). Application of Real-Time DEVS to Analysis of Safety-Critical Embedded Control Systems: Railroad Crossing Control Example. *The Society for Modeling and Simulation International*.
- Hrúz, B., & Zhou, M. (2007 ). *Modeling and Control of Discrete-event Dynamic Systems*. Londres: Springer-Verlag .
- Hu, X., Ganapathy, N., & Zeigler, B. (2006). Robots in the loop: supporting an incremental simulation-based design process. *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 2013 - 2018 Vol. 3 .
- Isermann, R. (June de 2007). Mechatronic systems—Innovative products with embedded control. *Science Direct Control Engineering Practice*, 14–29.
- Jovanov, E., & LeSueur, K. G. (2009). *Hardware-in-the-Loop Testing of Wireless Sensor Networks*. Huntsville, Alabama: ITEA Journal.
- Jung, D., & Tsiotras, P. (s.f.). *Modeling and Hardware-in-the-Loop Simulation for a Small Unmanned Aerial Vehicle*. Atlanta: Georgia Institute of Technology,.
- Khan, S. (2007). Supervisory Hybrid Control of a Micro Grid System. *IEEE Canada Electrical Power Conference*, 24.
- Kofman, E. (2003). *Discrete Event Based Simulation and Control of Continuous Systems*. Rosario, Argentina: Facultad de Ciencias Exactas, Ingeniería y Agrimensura Universidad Nacional de Rosario.
- Kofman, E. (Octubre de 2003). Quantized-state control: a method for discrete event control of continuous systems. *Lat. Am. appl. res., Bahía Blanca*, V 33(4), 399-406.
- Kofman, E., & Junco, S. (September de 2001 ). Quantized-state systems: a DEVS Approach for continuous system simulation. *Transactions of the Society for Computer Simulation International - Recent advances in DEVS Methodology--part I, Volume 18* (Issue 3), 123 - 132.

- LUNZE, J. (2009). *Handbook of Hybrid Systems Control*. New York: Cambridge University Press.
- Lygeros, J., & Johansson, H. K. (2003). Dynamical Properties of Hybrid Automata. *IEEE Transactions on automatic control*, 17.
- Michael, S. (2008). Assessment of performance and dependability in embedded control systems: Methodology and case study. *Control Engineering Practice*, 1307.
- Nica, M. (2008). Simulation of queues in manufacturing Systems. *ANNALS of the ORADEA UNIVERSITY.*, 1660.
- Nikolaïdou, M., & Dalakas, V. (2008). A SysML Profile for Classical DEVS Simulators. *Software Engineering Advances, 2008. ICSEA '08. The Third International Conference on* , 445 - 450 .
- Norman, G. (2000). *Administracion de produccion y operacion*. Mejico: international thomson editores.
- Oppenheim, A. (1998). Señales y sistemas. *Prentice Hall*, 957.
- Orooji, F., Sarjoughian, H., & Taghiyareh, F. (2010). Modeling & simulation of educational multi-agent systems in DEVS-suite . *Telecommunications (IST), 2010 5th International Symposium on* , 956 - 961 .
- Palaniappan, S. (2006). Application of the devs framework in construction simulation. *Proceedings of the 2006 Winter Simulation Conference*, 2086.
- Pepyne, D. (2000). Optimal Control of Hybrid Systems in Manufacturing. *Proceedings of the IEEE*, 1122.
- Proakis, J. (2007). Digital signal processing. *Pearson Prentice Hall*, 1084.
- Ramadge, P. (1989). Some Tractable Supervisory Control Problems for Discrete-Event Systems Modeled by Buchi Automata. *IEEE Transactions on automatic control*, 19.
- Ramadge, P. (1989). The Control of Discrete Event Systems. *IEEE Transactions on automatic control*, 98.
- Rezaie, K. (2009). Evaluating and Selecting Flexible Manufacturing Systems by Integrating Data Envelopment Analysis and Analytical Hierarchy Process Model. *Third Asia International Conference on Modelling & Simulation*, 464.
- Rocha, J. (1994). Task planning for flexible and agile manufacturing systems. *This work is partially supported by FLAD*, 112.
- Ryssel, U., Ploennigs, J., Kabitzsch, K., & Folie, M. (s.f.). *Generative Design of Hardware-in-the-Loop Models*. Dresden, Germany: Department of Computer Science Dresden University of Technology.
- the Object Management Group, Inc. (OMG). (2006). *OMG Systems Modeling Language (OMG SysML™)*. Atenas, Grecia: Standar.
- Von Neumann, J. (1966). Theory of Self-Reproducing Automata. *Fourth Lecture*, 87.
- Wainer, G. (2009). *Discrete Event Modeling and Simulation A practitioner's Approach*. Boca Raton, FL: Taylor & Francis Group, LLC.
- Wang, L., & Shen, W. (2007). *Process Planning and Scheduling for Distributed Manufacturing*. Ontario: Springer.
- Wei, L. (2007). Model-Based Design of a Dynamic Voltage Scaling Controller Based on Online Gradient Estimation Using Sim Events. *IEEE Conference on Decision & Control*, 6092.
- Winkler, D., & Gühmann, C. (s.f.). *Hardware-in-the-Loop simulation of a hybrid electric vehicle using Modelicar/Dymolar*. Berlin: Technical University of Berlin.
- Zeigler, B. (1993). Formalism and Methodology Unity of Conception Diversity of Application. *Winter Simulation Integration*, 579.
- Zeigler, B. (2000). *Theory of Modeling and Simulation Second Edition*. San Diego, CA: Academic Press.
- Zeigler, B. (2003). Today- recent advances in discrete event-based information technology. *Proceedings of the 11TH IEEE/ACM International Symposium on Modeling, Analysis and Simulation*, 14.
- Zhu, Y. (2009). Hardware-In-the-Loop Simulation of Pure Electric Vehicle Control System. *International Asia Conference on Informatics in Control*, 258.

## **Autorización y Renuncia**

Los autores autorizan a LACCEI para publicar el artículo en las actas de congresos. Ni los editores ni LACCEI no son responsables ni por el contenido ni por las implicaciones de lo que se expresa en el documento.